IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

BIAX CORPORATION,	§	
	§	
Plaintiff/Counterdefendant,	§	Civil Action No. 02-05CV-184-TJW
	§	
v.	§	
	§	The Honorable T. John Ward
INTEL CORPORATION	§	
	§	
and	§	**Markman Hearing Scheduled For**
	§	**November 17, 2006 at 9:00 A.M.**
ANALOG DEVICES, INC.	§	
	§	
Defendants/Counterclaimants.	§	

ANALOG DEVICES, INC.'S
RESPONSIVE BRIEF REGARDING CLAIM CONSTRUCTION FOR
U.S. PATENT NO. 5,765,037 PURSUANT TO PATENT LOCAL RULE 4-5(b)

TABLE OF CONTENTS

TABLE OF AUTHORITIES

Cases

Federal Statutes

I. INTRODUCTION

Analog Devices, Inc. ("ADI") respectfully submits this brief on the proper construction of asserted claims 1-12 of U.S. Patent No. 5,765,037 ("the '037 patent") and in response to plaintiff BIAX Corporation's ("BIAX") opening claim construction brief ("BIAX Br.").[1]

The '037 patent relates to a computer system design and method for "delayed branching," a concept that has been well-known in the computer industry since at least the 1970s. Indeed, several of the prior art references submitted to the Patent Office by BIAX's patent attorney described conventional delayed branching and "prepare-to-branch" mechanisms, including an article from 1982 entitled "Optimizing Delayed Branches." BIAX did not file its patent application until *1985*. Thus, the '037 patent must be construed narrowly such that it only encompasses what may be innovative in view of significant prior work by others in the field.

BIAX did not invent the concept of delayed branching. Based on its claim construction positions and infringement contentions against ADI, however, BIAX appears to assert that the '037 patent covers conventional delayed branching mechanisms that have been used in the computer industry since the 1970s and early 1980s. BIAX's position is untenable, and leaves no possibility of the '037 patent claims being valid, since conventional delayed branching systems were available to the public well before the filing date of BIAX's patent application in 1985. Thus, if the Court were to adopt BIAX's interpretations, the '037 patent is invalid.

In contrast, ADI's construction positions, which closely track the specifications, are more reasonable in view of the prior art systems. ADI's constructions are consistent with the fact that delayed branching and prepare-to-branch mechanisms were in the public domain long before BIAX's patent application was filed and the fact that the '037 patent describes a very specific computer system for performing delayed branching. Importantly, ADI's construction is also consistent with 1) BIAX's own use of the disputed claim language in the patent documents and 2) BIAX's explicit disclaimers made to distinguish its invention from prior systems. ADI does

[1] This brief is submitted separately because the '037 patent has only been asserted against ADI in this case.

not admit that, even as properly construed, the '037 patent is valid. What is certain, however, is that the BIAX's interpretation, which disregards the importance of the specification, file history, and prior art, is simply too broad given the limited nature of its invention.

II. THE TECHNOLOGY AT ISSUE

A. Brief Overview of the '037 Patent.

The '037 patent, entitled "System for Executing Instructions with Delayed Firing Times," describes a particular system "for effecting faster branch execution" by reordering instructions. (Ex. 1, abstract).[2] (As later discussed in Section II.B herein, reordering instructions for increased performance of branch instructions was well known in the prior art before BIAX filed its patent application in 1985.)

The '037 patent describes that a computer program is compiled to produce "object code"[3] which is in the form of "basic blocks."[4] (Ex. 1, col. 4, ll. 37-41; col. 6, ll. 52-55). Each basic block may contain several instructions, but has only one entry point and only one exit point – no more than one of the instructions may perform a branch operation, and that instruction must be the last instruction of the block. (Ex. 1, col. 6, ll. 52-55; col. 6, ll. 66 to col. 7, ll. 1).

In computing, program instructions are generally "fetched" (*i.e.*, retrieved from memory) and processed sequentially by a computer. "Branch instructions" are a certain type of instruction that tells the computer to skip a portion of the instructions and jump – or branch – to another section of the instructions out of sequence.[5] The processing of branch instructions, however, slows down the execution of programs by breaking the flow of fetching and processing.

---

[2] The exhibits referenced in this brief as "Ex. __" are attached to the Declaration of Jim Taylor In Support Of Analog Devices Inc.'s Responsive Brief Regarding Claim Construction For U.S. Patent No. 5,765,037, filed herewith.

[3] "Object code" refers to the code produced by a compiler from the source code, usually in the form of machine language that a computer can execute directly.

[4] "Basic blocks" means "compiler output consisting of groups of instructions with only one branch at the end of each block." (*See* Section III.B.9).

[5] "Branch instructions" mean "a control flow instruction that can change the sequence in which the program executes instructions. A branch instruction is executed by a separate branch execution unit." (*See* Section III.B.4).

As explained in the '037 patent, a branch instruction involves three actions or steps: (a) evaluation of the "branch condition";[6] (b) generation of the "branch target instruction address";[7] and (c) placing the target instruction address into the "next instruction fetch register."[8] (Ex. 1, col. 36, ll. 31-38). Because of the time that it can take to execute steps (a) through (c) of a branch instruction, the fact that the branch instruction is the last instruction in the basic block can result in a delay between the completion of the instructions in that basic block and the fetching of the instructions at the branch target address. (Ex. 1, col. 38, ll. 21-32).

The '037 patent describes a system that causes the branch instruction to begin execution – steps (a) and (b) – before the computer reaches the end of a basic block, but delays completion of the branch – step (c) – until the last non-branch instruction is being processed. (Ex. 1, abstract). This allows non-branch instructions from the basic block to be fetched and/or processed between the time that the branch begins execution and the time that it completes execution. The '037 patent describes this as "delayed branching." (Ex. 1, col. 37, ll. 9-10).

The '037 patent describes the use of "TOLL software" which adds an "instruction firing time"[9] to the branch instruction. (Ex. 1, col. 6, ll. 33 to col. 14, ll. 21). The instruction firing time identifies a "time of execution"[10] of the branch instruction, which is a "variable number of instruction cycles"[11] prior to a time of execution of a last to be executed non-branch instruction. (Ex. 1, col. 4, ll. 37-60). The '037 patent also describes "Processor Elements" for processing non-branch instructions. (Ex. 1, col. 40, ll. 20 to col. 45, ll. 33). The Processor Elements – which the '037 patent describes as being identical, homogenous, context-free devices that can

[6] "Branch condition" refers to the status, or condition, of the branch instruction.

[7] "Branch target instruction address" refers to the instruction at the address targeted by the branch.

[8] "Next instruction fetch register" refers to a register (*i.e.*, a special storage area where data is stored before it can be processed) from which the next instruction is fetched.

[9] "Instruction firing time" means "a line of computer code providing the specific time when an instruction is to be executed." (*See* Section III.B.1).

[10] "Time of execution" means "the time at which an instruction begins processing and execution." (*See* Section III.B.8).

[11] "Variable number of instruction cycles" means "a delay period lasting a number of instruction cycles that changes based on the assignment of the branch instruction to an earliest possible instruction firing time." (Section III.B.3).

execute all instructions except branch instructions – are shown below (circled) in Fig. 6 of the '037 patent. (Ex. 1, col. 41, ll. 5-13; col. 41, ll. 22-24; col. 4, ll. 17-28; and col. 15, ll. 9-24)

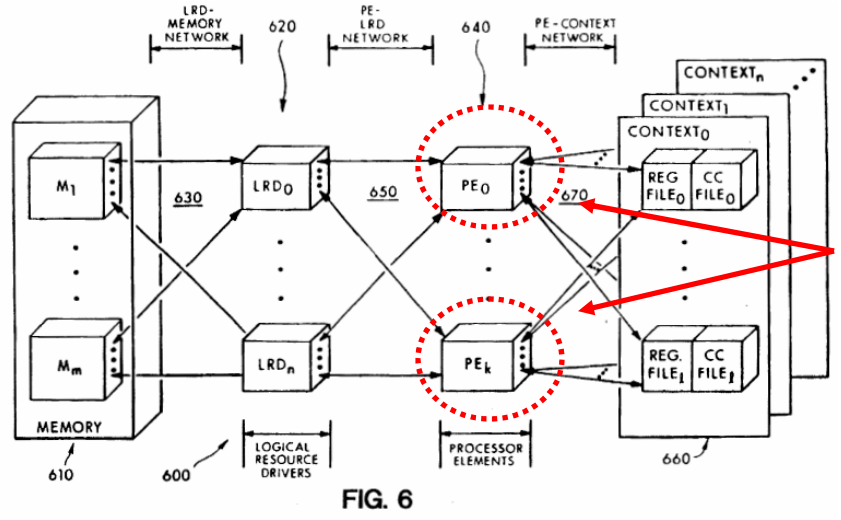

FIG. 6

The '037 patent further describes a "Branch Execution Unit" that responds to the instruction firing time and completes the execution of the branch instruction no later than the same time as the processing element is processing the last to be executed non-branch instruction, so that the execution of the branch instruction occurs in parallel with execution of the non-branch instructions. (Ex. 1, col. 36, ll. 19 to col. 40, ll. 18). The Branch Execution Unit is part of the "Logical Resource Drivers" (LRD) shown in Fig. 6 above. (Ex. 1, col. 28, ll. 64-66). The Branch Execution Unit (BEU) is illustrated in Fig. 19 of the '037 patent.

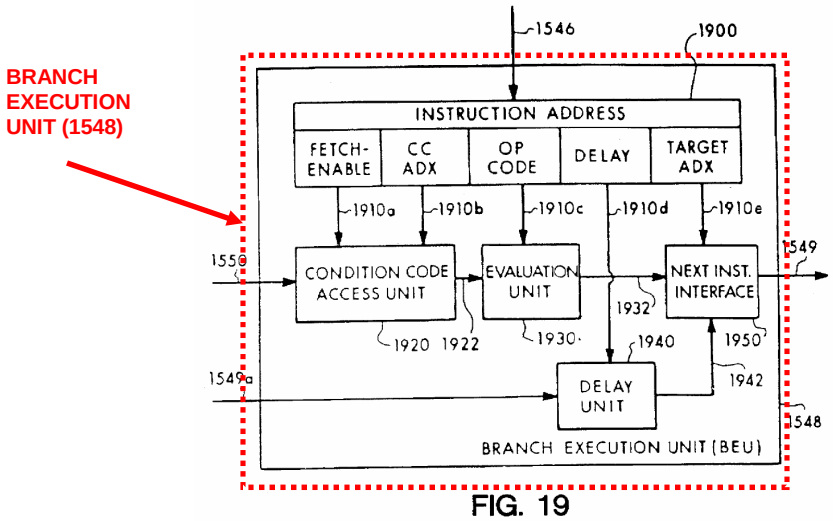

FIG. 19

The '037 patent states that the Processor Elements execute all non-branch instructions, while the Branch Execution Unit executes all branch instructions. (Ex. 1, col. 12, ll. 37-40).

The '037 patent explains that the BIAX system uses "delayed branching to guarantee program correctness." (Ex. 1, col. 37, ll. 9-10). As shown below in Table 6 of the '037 patent, branch instruction I5 is scheduled by its instruction firing time to begin execution in T17. The last to be executed non-branch instruction, I3, however, is scheduled by its instruction firing time for T18. (Ex. 1, col. 12, ll. 52-67).

TABLE 6

Logical Processor Number	T16	T17	T18
0	I0	I2	I3
1	I1	—	
2	I4	—	—
3	—	—	—
BEU	—	I5 (delay)	—

If branch instruction I5 were to complete execution in T17, then the system would branch or jump to another instruction in T17. In T18, the computer would process the instruction at the address targeted by the branch – the branch target address. The next instruction in sequence, instruction I3, would be lost and the program would not execute correctly. To execute the program correctly, the branch instruction begins in T17 but does not complete in T17. (Ex. 1, col. 12, ll. 63-67). Even though the branch is processed in T17, its execution is delayed until after the last instruction, I3, has been executed. (*Id.*).

The Branch Execution Unit processes the first three steps of the branch instruction in T17, but a delay value from the branch instruction is sent to a delay unit to cause a delay during T18 before allowing the next instruction interface to start the instruction at the target address in T19. (Ex. 1, col. 12, ll. 63-67). The computer program is executed during parallel processing by beginning to process the branch at a time when other instructions are processed, but delaying the execution of the branch until after execution of all the other instructions that precede the branch in the program. (*Id.*).

B. Delayed Branching Mechanisms Have Been Known Since The 1970s.

BIAX does "not claim to be the first to have desired to execute instructions in parallel in the shortest amount of time. This is indeed a common goal of many researchers and engineers." (Ex. 6, Serial No. 480,841, Second Prelim. Amend., Aug. 1, 1995, at 14).

Prior to BIAX's filing of its original patent application in October 1985, there was extensive prior art on delayed branching and prepare-to-branch mechanisms by others, including Herbert Schorr (IBM, 1971), Thomas R. Gross and John L. Hennesey (Stanford University, 1982), and Honesty Young and James Goodman (University of Wisconsin, 1984).

ADI believes that a short discussion of these three prior art references, and the nature of delayed branching and prepare-to-branch mechanisms, will aid the Court in understanding the context of the asserted claims of the '037 patent, since this prior art places limits on the scope and interpretation of the claims.

1. Schorr, "Design Principles for a High Performance System," Symposium on Computers and Automata (1971)

In 1971, Herbert Schorr of IBM published an article entitled "Design Principles for a High Performance System" (Ex. 3), which described a technique referred to as "branch anticipation" in which delayed branches were used to reduce performance losses caused by conditional branches. This technique was combined with rearranging code to take advantage of parallelism and to limit delays. (Ex. 3, at 167). (The Schorr article was not shown to the Patent Office by BIAX, but is now before the Patent Office in a reexamination request proceeding involving the '037 patent.)

The Schorr article described replacing the traditional branch at the end of basic block by a "prepare-to-branch" (referred to as "branch" in Fig. 5 of the Schorr article) operation placed earlier in the basic block and an exit instruction in the normal branch position at the end of the basic block. (Ex. 3, at 169-170, Fig. 5). The "prepare-to-branch" or "branch" instruction performed the first two steps of the branch (*i.e.*, condition determination and branch address

generation), and the "exit" instruction performed the final step of the branch (*i.e.*, placement of target address in next instruction fetch register). (Ex. 3, at 170, Fig. 5(b)).

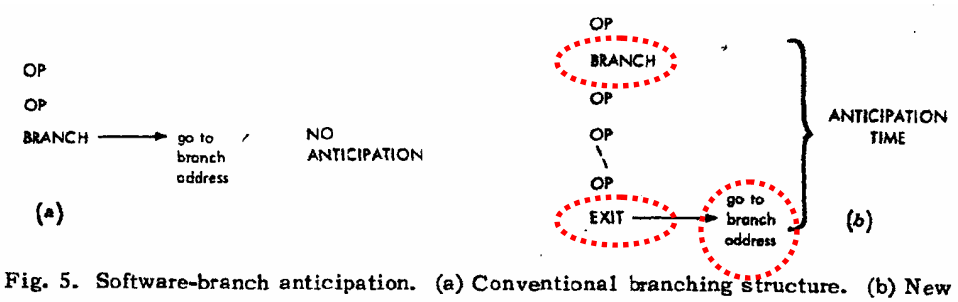

Fig. 5. Software-branch anticipation. (a) Conventional branching structure. (b) New branching structure.

In the Schorr article, a variable number of instructions ("OP") were scheduled between the "prepare-to-branch" and "exit" points. (Ex. 3, at 170, Fig. 5(b)). Schorr introduced delay in execution of a conditional branch by the relative placement of the prepare-to-branch instruction and the exit instruction. (Ex. 3, at 169). This organization allowed the branch execution to occur concurrently with non-branch instructions of the block. (Ex. 3, at 170, Fig. 5(b)).

2. Gross et al., "Optimizing Delayed Branches," IEEE, 15th Annual Workshop on Microprogramming (1982)

In 1982, Thomas R. Gross and John L. Hennessey of Stanford University published an article entitled "Optimizing Delayed Branches" (Ex. 4), which discussed code optimization techniques for pipelined architectures, and in particular, discussed delayed branches as a code optimization technique. (The Gross article was cited during the prosecution of the '037 patent).

In a "pipelined" computer, instructions are batched and executed in a series of steps, with each step being performed by a different pipeline stage. Once one pipeline stage is through performing its step, it passes the result to the next pipeline stage and accepts the next instruction in the sequence. As a result, if several instructions are being processed in the pipeline at the same time, each being in a different stage of processing, a branch instruction at the end of a basic block can cause a problem because the processor does not know which instructions to fetch until the branch is executed, but by the time the branch has reached the point in the pipeline at which this determination is made, a number of instructions following the branch may have already been

fetched in the pipeline. The number of instructions following a branch that will be fetched into the pipeline is referred to as the "branch delay" of a pipelined processor.

The Gross article described that a MIPS ("Microprocessor without Interlocked Pipeline Stages") processor that employed delayed branches to avoid a complex instruction prefetch and to simplify pipeline design. (Ex. 4, at 114-15). The Gross article described a delayed branch algorithm wherein non-branch instructions are moved to positions after the branch instruction, thereby rescheduling the branch instruction. (Ex. 4, at 116). The Gross article described three techniques for dealing with delayed branches: 1) move n instructions from before the branch to after the branch; 2) duplicate the first n instructions from the branch target and branch to the nth instruction after the target instead; and 3) move the next n sequential instructions to immediately after the branch. (*Id.*). Using these techniques, the Gross article reported that higher performance could be achieved. (Ex. 4, at 120).

3. Young et al., "A Simulation Study of Architectural Data Queues and Prepare-to-Branch Instruction," IEEE International Conference on Computer Design: VLSI in Computers (1984)

In 1984, Honesty Young and James Goodman published an article entitled "A Simulation Study of Architectural Data Queues and Prepare-to-Branch Instruction" (Ex. 5), which described the development of a computer system at the University of Wisconsin called the PIPE ("Parallel Instruction with Pipelined Execution") computer. (The Young article was not shown to the Patent Office by BIAX, but is now before the Patent Office in a reexamination request proceeding involving the '037 patent.)

The Young article described a "prepare-to-branch" instruction having a built-in delay field. (Ex. 5, at 545). The Young article discussed the addition of a branch count (BC) field to the prepare-to-branch (PBR) instruction which indicates when the branch will be executed. (*Id.*). The branch count is shown to be a variable number. (Ex. 5, at 547). The Young article also describes an optimizing compiler and code scheduler which takes advantage of the prepare-to-branch instruction. (Ex. 5, at 545-47). Young stated that the prepare-to-branch instruction is a mechanism to decrease the penalty incurred by conditional branches. (Ex. 5, at 545).

It is against this background – including the Schorr, Gross, and Young prior art – that the invention of the '037 patent, and the language used to claim that invention, must be construed.

C. After Filing Its Patent Application In 1985, BIAX's Delayed Branching Claims Did Not Issue Until Over 12 Years Later, In 1998.

BIAX filed its original patent application on October 31, 1985.[12] Out of this initial application, BIAX filed and prosecuted a number of continuation applications over a period of 15 and a half years, with the last patent, the '313 patent, issuing in June 2001. One of the continuation applications, filed in June 1995, led to the issuance of the '037 patent, in June 1998. Beginning with its initial patent application in 1985, BIAX sought to prosecute claims for reordering and executing branch instructions – the delayed branching claims. (Ex. 2, Serial No. 794,221, *see* claims 71 and 72). The delayed branching claims took over 12 years to issue.

D. The Asserted Claims Of The '037 Patent.

BIAX has asserted claims 1-12 of the '037 patent against ADI. All twelve of the asserted claims are independent claims, and thus each of those independent claims must be construed. Generally, the asserted claims of the '037 patent can be categorized as follows:

1. Apparatus Claims. Asserted claims 1-6 are apparatus claims in "means-plus-function" format. BIAX and ADI agree that the elements of claims 1-6 should be construed as means-plus-function language pursuant to 35 U.S.C. § 112, ¶ 6.

2. Method Claims. Asserted claims 7-12 are method claims. These claims are directed to machine implemented methods. The method limitations of claims 7-12 parallel the apparatus limitations of claims 1-6.

3. "Multiple Users" Claims. Half of the asserted claims – claims 2, 4, 6, 9, 11, 12 – are directed to multiple users (*i.e.*, persons). In particular, these claims are directed to

[12] As noted in the Defendants' Joint Brief filed herewith, over two and a half years after BIAX filed its original application, BIAX's new patent attorney submitted a "substitute specification" that significantly enlarged the scope of the original disclosure. For the purposes of claim construction, BIAX should be estopped from relying on the substitute specification to support its proposed definitions wherever the substitute specification departs from or is inconsistent with the original. (*See* Section III of Defendants' Joint Claim Construction Brief).

systems and machine methods for executing branches in basic blocks in "a plurality of programs utilized by a *number of users*." (emphasis added).

BIAX refused to limit the number of '037 patent claims asserted against ADI. Rather, BIAX's counsel told ADI that they preferred to see how this Court construed the claims before making any decisions on whether to limit the number of claims asserted. BIAX's refusal to reduce the number of asserted claims to a reasonable number has resulted in a large volume of claim terms over which the parties disagree. Given the volume of claim terms in dispute in the '037 patent, it is not feasible for ADI to fully brief each one of them. Instead, ADI will focus on those terms which it believes are most relevant to a fair and speedy resolution of the case.

III. THE PROPER CONSTRUCTION OF CLAIM LANGUAGE IN DISPUTE

This brief does not respond point by point to BIAX's opening brief because, in ADI's view, the BIAX brief does not necessarily address the key terms. Instead, given all the claims asserted of the '037 patent by BIAX, ADI focuses on the most significant terms.

A. Legal Framework For Claim Construction.

ADI incorporates herein by reference Section IV of the Defendants' Joint Brief, entitled "Law Governing Claim Construction." In addition, ADI adds one following thought:

Understanding the "invention" is essential to an accurate delineation of a patent holder's right to exclude others from the field of her invention:

> Ultimately, the interpretation to be given a term can only be determined and confirmed with a full understanding of what the inventors actually invented and intended to envelop with the claim. The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction.

Renishaw PLC v. Marposs Societa' Per Azioni, 158 F.3d 1243, 1250 (Fed. Cir. 1998) (citation omitted). The claims must be read in the context of the problems solved, and with a common sense approach to the objectives of the invention. *See Pall Corp. v. Micron Separations, Inc.*, 66 F.3d 1211, 1217 (Fed. Cir. 1995), *cert. denied*, 520 U.S. 1115 (1995).

B. Key Terms And Phrases In Dispute.

1. "Instruction firing time," "Instruction firing time information," and "Firing time information" (claims 1, 2, 5, 6, 10, 11)

Phrases to be Construed	ADI's Construction	BIAX's Construction
"instruction firing time"	The specific time when an instruction is to be executed.	*See* definition of "instruction firing time information."
"firing time information" "instruction firing time information"	A line of computer code providing the specific time when an instruction is to be executed.	Information pertaining to the time when an instruction is scheduled to be executed.

Claim construction begins with the language of the claims. Examining the language of the asserted claims of the '037 patent, however, does not reveal any information that enlightens the construction of the limitations "firing time information," "instruction firing time information," or "instruction firing time" – these are terms coined by the inventors and which have no ordinary meaning. As far as ADI can tell, these phrases do not appear in any dictionary, technical or otherwise. Furthermore, other than appearing in the BIAX patents, the phrase "instruction firing time" does not appear in any other U.S. patent, except for one, U.S. Patent No. 5,613,080, which simply mentions BIAX's '755 patent and "instruction firing time" in a section entitled "Description of Related Art."

Turning to the specification of the '037 patent, the proper construction begins to come into clear focus. It is clear from the specification that "instruction firing time" provides the time-driven control of the invention. (Ex. 1, col. 16, ll. 46-47).

The proper construction of "instruction firing time" is "the specific time when an instruction is to be executed." In addition, the proper construction of the related phrases "firing time information" and "instruction firing time information" (which BIAX concedes should be construed together) is "a line of computer code providing the specific time when an instruction is to be executed." These constructions are consistent with BIAX's 1) own use of this language in its patents and 2) explicit disclaimers made to distinguish its invention from prior systems.

As an initial matter, the specification of the '037 patent states: "The system and method of ***the present invention*** are described in the following drawing and specification." (Ex. 1, col. 4, ll. 61-62). This is important because references to a particular feature as part of "the present invention" limit the claims to structures and methods utilizing that feature. *See, e.g, Microsoft Corp. v. Multi-Tech Sys., Inc.*, 357 F.3d 1340, 1348 (Fed. Cir. 2004) (particular embodiment defined the invention based on "clear statements in the specification that the invention ('the present system') is directed to communications 'over a standard telephone line'"); *Watts v. XL Sys., Inc.*, 232 F.3d 877, 883 (Fed. Cir. 2000). ("[T]he specification actually limits the invention to structures that utilize misaligned taper angles, stating that '[t]he present invention utilizes [the varying taper angle] feature.'").

Both the specification and prosecution histories of the BIAX patent family describe the instruction firing time as the ***specific time*** when an instruction is to be executed. For example, during prosecution of one of its patent applications, BIAX told the Patent Office that:

> ***[T]he present invention*** explicitly extends each instruction with a logical processor number as well as providing ***a specific firing time*** for the instruction.

(Ex. 7, Serial No. 794,221, Pet. for Advance. Exam. (Oct. 31, 1985), at 11) (emphasis added). Therefore, the constructions of "instruction firing time," "instruction firing time information," and "firing time information" must be construed in light of this statement in the prosecution file history. *See Watts, 232 F.3d at 883* (applying patentee's arguments distinguishing the prior art to all claims in the patent).

BIAX's proposed construction ambiguously defines "instruction firing time information," "firing time information" and "instruction firing time" (in the context of the '037 patent) as "information pertaining to the time when an instruction is scheduled to be executed." BIAX emphasized during prosecution of its patents that "providing ***the*** firing time" is "substantially different" than the prior art, which described a relative range of times for executing instructions:

> ***Freiman et al do not provide firing times but only the notion of placing a range*** about the times during which an instruction can be performed. ***This is substantially different than the requirement of providing the firing time*** for each

> instruction, in particular, in a multiple program computer configuration such as that of claim 7.

(Ex. 8, Serial No. 794,221, Resp. to Office Action (Dec. 22, 1988), at 10) (underlined emphasis in original; bold emphasis added). In addition, BIAX argued to the Patent Office:

> ***In particular, Freiman et al merely provide the vaguest reference to the "firing time" in column 3, at lines 1-5. He only provides the so-called latest and earliest times during which an operation can be performed. This is not equivalent to providing <u>the</u> instruction firing time itself.*** That is, the time when an instruction is scheduled to be executed. (Note that this may be neither the earliest nor the latest time when the operation can be performed, depending upon the other instructions and resources available to the system.)

(Ex. 9, Serial No. 372,247, Resp. to Office Action (Jul. 27, 1990), at 17) (emphasis added).

In addition, BIAX's proposed construction – "*information pertaining to* the time when an instruction is scheduled to be executed" – would render the claims of the '037 patent invalid, because systems for reordering and executing branch instructions using "information" were in the public domain well before BIAX filed its patent application in 1985. For example, the Schorr, Gross, and Young prior art discussed above cannot be distinguished from the '037 patent under BIAX's construction, because each of those references described reordering and executing branch instructions using "information" pertaining to a time when instructions were executed.

BIAX's construction is directly contradicted by statements that it made during the prosecution history, in which BIAX distinguished its invention from the prior art by stating that the instruction firing time ***was physically attached*** to each instruction:

> In other words, McDowell attempts to schedule instructions with differing length execution times so that they can execute concurrently with a minimum amount of hardware support. To the contrary, the present invention does not require this third component. ***The use of the logical processor number and the instruction firing time that <u>are physically attached</u> to each instruction preclude the need for this phrase***.

(Ex. 7, Serial No. 794,221, Pet. for Advance. of Exam. (Oct. 31, 1985), at 8).

ADI's construction of "firing time information" and "instruction firing time information" as "a line of computer code providing the specific time when an instruction is to be executed" is consistent with BIAX's statement that the instruction firing time is physically attached (*i.e.*, appended as a line of computer code) to each instruction. If the instruction firing time was not

physically attached to each instruction, BIAX's "invention" would be no different than the prior art systems available before 1985, which also reordered instructions.

"Claims may not be construed one way in order to obtain their allowance and in a different way against accused infringers." *Southwall Techs., Inc. v. Cardinal IG Co.*, 54 F.3d 1570, 1576 (Fed. Cir. 1995). Furthermore, explicit statements or arguments to the Patent Office during prosecution distinguishing the invention from prior art may function as a disclaimer for claim construction purposes. *See, e.g., Alpex Computer Corp. v. Nintendo Co. Ltd.*, 102 F.3d 1214, 1220-22 (Fed. Cir. 1996) (finding patentee effectively disclaimed any video display system based on shift registers where, during patent prosecution, patentee distinguished the claimed invention from the prior art).

BIAX's broad construction of the phrases "instruction firing time," "firing time information" and "instruction firing time information" attempts to capture a certain meaning that was expressly disclaimed during prosecution. BIAX is not entitled to a "mulligan that would erase from the prosecution history the inventor's disavowal of a particular aspect of a claim term's meaning" to achieve a strategic objective during litigation. *Hockerson-Halberstadt, Inc. v. Avia Group Int'l, Inc.*, 222 F.3d 951, 957 (Fed. Cir. 2000).

Accordingly, ADI's construction should be adopted by this Court.

2. "Scheduling processing of said branch instruction[s]" (claims 8, 9) and "Adding information to said branch instruction[s]" (claims 7, 12)

Phrases to be Construed	**ADI's Construction**	**BIAX's Construction**
"scheduling processing of said branch instruction[s]"	Adding to each of the branch instructions computer code that assigns an instruction firing time.	Does not require construction because it has the ordinary meaning of the words in the term and any terms needing construction are construed elsewhere herein.
"adding information to said branch instruction[s]"	Adding to each of the branch instructions computer code that assigns an instruction firing time.	Does not require construction because it has the ordinary meaning of the words in the term and any terms needing construction are construed elsewhere herein.

The proper construction of "scheduling processing of said branch instruction"[13] (claims 8 and 9) and "adding information to said branch instructions"(claims 7 and 12) is "adding to each of the branch instructions computer code that assigns an instruction firing time." BIAX does not offer any construction for these phrases, apparently preferring to leave the phrases ambiguous.

According to the Federal Circuit: "Ultimately, the interpretation to be given a term can only be determined and confirmed with a full understanding of what the inventors *actually* invented and intended to envelop with the claim. *The construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction.*" *Phillips v. AHW Corp.*, 415 F.3d 1303, 1316 (Fed. Cir. 2005). (emphasis added). Therefore, "[i]t is entirely appropriate for a court, when conducting claim construction, to rely heavily on the written description for guidance as to the meaning of the claims." *Phillips*, 415 F.3d at 1317. Where, as here, the specification consistently equates the claim language with the *only* disclosed embodiment, and every example of the disputed claim language in the specification consistently points to an implicit definition, *that definition must control*. *See Phillips*, 415 F.3d at 1321.

The specification of the '037 patent – which is the "single best guide" to the meaning of the disputed claim language, *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996) – makes clear that the phrases "scheduling processing" and "adding information" must be construed to mean "adding to each of the branch instructions computer code that assigns an instruction firing time":

- "***The instruction firing time (IFT) provides the 'time-driven' feature of the present invention.***" ('037 patent, col. 12, ll. 18-20).
- "***[T]he object code instructions are <u>assigned</u> 'instruction firing times' (IFTs)...")*** (Ex. 1, '037 patent, col. 11, ll. 66 to col. 12, ll. 15.)
- "***The latest time determines the actual firing time <u>assigned</u> to the instruction.***" (Ex. 1, '037 patent, col. 10, ll. 46-47.)

[13] Although the phrase "scheduling processing of said branch instruction[s]" is contained within claims 3 and 4, the disputed language in those claims should be construed pursuant to 36 U.S.C. § 112, § 6. (*See* Section III.C.2).

- The TOLL software "***assigns* instruction firing times (IFTs)** on the basis of this resource usage (e.g., Table 4), reorders the instruction stream based upon these firing times (e.g., Table 5) and assigns logical processor numbers (LPNs) (e.g.. Table 6) as a result thereof." (Ex. 1, '037 patent, col. 10, ll. 46-47.)

- "The ***assignment* of the firing time** (and in the illustrated embodiment, the logical processor number) is dependent on how the instruction stream accesses these resources." (Ex. 1, '037 patent, col. 19, ll. 11-14.)

- "When all of the instructions of a basic block have been ***assigned* firing times**, the maximum firing time of the current basic block (the one the branch is a member of) is used to update all resources load and free times (to this value)." (Ex. 1, '037 patent, col. 19, ll. 11-14.)

- "The present invention statically determines at a very fine level of granularity, the natural concurrencies in the basic blocks (BBs) of programs at essentially the object code level and ***adds*** a logical processor number (LPN) ***and an instruction firing time to each instruction in each basic block in order to provide a time driven decentralized control***." (Ex. 2, Original Specification, at 28, ll. 8-18.)

- "***The extended intelligence information involving the logical processor number (LPN) and the instruction firing time (IFT) is added to each instruction of the basic block***..." (Ex. 2, Original Specification, at 46, ll. 28-32.)

- "***Hence, under the teachings of the present invention, the added intelligence as shown in Figure 4, is based upon detected natural concurrencies within the object cods. The detected concurrencies are analyzed by TOLL which logically assigns individual logical processor elements (LPNs) to process the instructions in parallel, and assigns unique firing times (IFT) so that each processor element (PE) for its given instruction will have all necessary resources available for processing according to its instruction requirement***." (Ex. 2, Original Specification, at 53, ll. 13-23. *See* '037 patent, col. 16, ll. 29-38.)

The Federal Circuit has stated that "claims may be no broader than the supporting disclosure, and therefore that a narrow disclosure will limit claim breadth." *Gentry Gallery, Inc. v. The Berkline Corp.*, 134 F.3d 1473, 1480 (Fed. Cir. 1998). "[W]here there is an equal choice between a broader and a narrower meaning of a claim, and there is an enabling disclosure that indicates that the applicant is at least entitled to a claim having the narrower meaning, we consider the notice function of the claim to be best served by adopting the narrower meaning." *Athletic Alternatives, Inc. v. Prince Mfg., Inc.*, 73 F.3d 1573, 1581 (Fed. Cir. 1996)

The prosecution history also makes clear that the phrases "scheduling processing" and "adding information" must be construed to mean assigning an instruction firing time:

> As the Examiner correctly notes, ***these claims*** are directed to branch instruction execution in a particular environment wherein, inter alia, ***firing time information***

> ***is <u>added</u> to a branch instruction and the branch instruction can begin execution*** . . . a number of cycles prior to the execution time of the last non-branch instruction and at the same time complete execution no later than the completion of the last non-branch instruction.

(Ex. 10, Serial No. 480,841, Resp. to Office Action, at 13) (emphasis added).

Based on BIAX's explicit statement above that ***"<u>these claims</u>"*** – the delayed branching claims that issued in the '037 patent – related to a "particular environment" in which "firing time information is added to a branch instruction," the claim phrases "scheduling processing" and "adding information" must involve adding firing time information. BIAX continued:

> Referring to the substitute specification, the portions of the specification which relate to and describe a claimed invention are primarily ground starting on page 14, beginning at line 5, continuing to the top of page 24. In addition, the description relating to the branch execution unit (BEU), starting at page 68, line 28, and continuing to page 76, line 5, is also important. ***The first referenced material described how the branch instruction is modified to include <u>instruction firing time</u> and a delay, the delay being a variable number.***
>
> . . .
> ***[T]he "TOLL" software 110 creates the instruction firing times and adds those times to the branch (and other) instructions.*** The branch instruction also receives, in addition, a delay which can prevent it from actually completing execution until some time later. ***Thus, [at] the time indicated by the instruction firing time, the branch instruction, the branch instruction is loaded into the branch instruction [sic] unit;*** however, the completion of its operation is prevented by the "delay" (which may be "zero") as illustrated both in table 6 on page 23 and in connection with the description of the branch execution in Figure 19. In Figure 19, the delay is controlled by the delay unit 1940. While ***the actual value of either the instruction firing time or the delay will be a function of the elements of the basic block with which they are associated***, it is clear from the substitute specification starting, for example, at page 14, line 5, that the software 110 is designed to find the earliest of the possible instruction firing times for all instructions, including branch instruction [sic] (for example, instruction I5 in the example described in Tables 3, 4, 5, and 6). The particular method illustrated in the application for identifying earliest resources time is described in connection with, for example, Figures 10 and 11. ***These figures, which describe the analysis and determination of the intelligence of means 120, 130, respectively, illustrate to one of ordinary skill in the art, without any undo experimentation, how to build the instruction firing time methodology for a branch instruction and how to effect the delay in order that the branch instruction will execute no later than the last non-branch instruction.***

(Ex. 10, Serial No. 480,841, Resp. to Office Action, at 14-15) (emphasis added).

The above prosecution statements by BIAX that an "instruction firing time" is added to branches by the TOLL software; that loading begins "at the time indicated by ***<u>the</u>*** instruction

firing time;" that an "instruction firing time" has an "actual value" determined by the instructions in the basic block; and that one of ordinary skill in the art is expected "to build the instruction firing time," all confirm, that "information" in the claims refers to an "instruction firing time" and that "scheduling" refers to adding an instruction firing time to an instruction.

BIAX also told the Patent Office:

> Further, the delay unit which is illustrated in Figure 19, and which is described in detail in the corresponding text, illustrates how branch instruction takes place, and describes, with the TOLL philosophy and the methodology noted Figures 8, 9, 10, and 11, ***the [branch] instruction will obtain an instruction firing time as early as possible in accordance with the invention*** and that the delay will be what it must be, that is, a variable number of cycles, before the execution time of the last non-branch instruction. The is the entire "TOLL" philosophy and is built into the description at every opportunity. The description thereby enables the invention claimed herein and the drawings provide the necessary support for each feature.

(Ex. 10, Serial No. 480,841, Resp. to Office Action, at 15) (emphasis added).

The above prosecution statement by BIAX again confirms that an "instruction firing time" is added to branch instructions "in accordance with the invention." Statements noting that a particular feature is present "according to the invention" indicate that feature is required by the claimed invention. *Gaus v. Conair Corp.*, 363 F.3d 1284, 1290 (Fed. Cir. 2004) (description of a protective circuit as "according to the invention" held to "demonstrate[] that the invention itself requires that the user be completely protected from shock").

Although BIAX has not done so yet, to the extent it argues in its reply that the doctrine of claim differentiation presumes that claims should be given different scope, "[t]he presumption that separate claims have different scope 'is a guide, not a rigid rule.'" *See ATD Corp. v. Lydall, Inc.*, 159 F.3d 534, 541 (Fed. Cir. 1998). Importantly, "[t]he doctrine of claim differentiation can not broaden claims beyond the scope that is supported by the specification." *Id.*; *see also Multiform Desiccants, Inc. v. Medzam, Ltd.*, 133 F.3d 1473, 1480 (Fed. Cir. 1998) ("claim differentiation can not broaden claims beyond their correct scope").

BIAX's position here – that the claim phrases "scheduling processing of said branch instruction" and "adding information to said branch instructions" do not need to be construed – ignores that fact that systems for reordering and executing branch instructions by "scheduling

processing" and "adding information" were well known before BIAX filed its patent application in 1985. The Schorr, Gross, and Young prior art all described reordering and executing branch instructions by scheduling processing and adding information to the branch instructions.

In the '037 patent, *all* branch instructions are reordered by adding computer code to the branch instruction that assigns an instruction firing time. That was the purpose of the '037 invention. That was the improvement that the Patent Office allowed. That was the only embodiment described in the patent. That is all the claims can be construed to cover.

3. "Variable number of instruction cycles" (claims 1-12)

Phrases to be Construed	**ADI's Construction**	**BIAX's Construction**
"variable number of instruction cycles"	A delay period lasting a number of instruction cycles that changes based on the assignment of the branch instruction to an earliest possible instruction firing time.	Does not require construction because it has the ordinary meaning of the words in the term and any terms needing construction are construed elsewhere herein. *See* "instruction cycle."
"instruction cycles"	A time interval for full processing and execution of one or more instructions.	The period of time between the issuing of two successive instructions.

The claim phrase "variable number of instruction cycles," which is in each of asserted claims 1-12, should be construed to mean "a delay period lasting a number of instruction cycles that changes based on the assignment of the branch instruction to an earliest possible instruction firing time." BIAX does not offer a proposed construction for this phrase.

Examining the language of the claims does not reveal any information that enlightens the meaning of the phrase "variable number of instruction cycles." Even the Patent Office had difficulty understanding the term "variable," telling BIAX that: "It is not clear in what sense 'variable' is meant, because a delay is a definite number." (Ex. 11, Serial No. 480,841, Office Action (Jan. 10, 1997), at 2). The fact that the term "variable" is difficult to understand is not surprising, because the specification only mentions the word "variable" once, in the "Summary of the Invention" section:

> The method and apparatus feature receiving the program for determining the branch instruction within each basic block of the program and adding information to the branch instruction, the information identifying, in at least one instance, a time of execution of the branch instruction which is a ***variable number*** of instruction cycles prior to a time of execution of a last to be executed instruction of the basic block.

(Ex. 1, col. 4, ll. 41-48).

Notably, the above paragraph was not in the original specification filed in 1985, but was added during prosecution of the '037 patent. Thus, there is a real question whether the asserted claims are even enabled by the specification.

Still, the prosecution history of the '037 patent provides the best evidence of what BIAX meant by "variable number of instruction cycles," because BIAX had to respond to the Patent Office's rejection of the claims on the basis of indefiniteness. BIAX responded as follows:

> Further, with particular regard to claims 74-85, and the Examiner's questions, ***the "variable number" depends upon the circumstances of the order of the instructions and is determined at the time typically prior to reordering of the instructions.*** This process is particularly examined in detail in the specification, and in particular, in that portion of the specification dealing with the branch instructions.
>
> Further, with respect to claims 74-85, ***while the Examiner is correct that the delay is a "definite number," the value of the delay can vary depending upon the circumstances of the instruction flow. It is this variable value of the delay to which the "variable number" refers.*** For example, compiler system as described in the application determines the delay value which assures a completion of the branch instruction execution prior to the completions of the non-branch instruction execution as required by the claim.

(Ex. 12, Serial No. 480,841, Resp. to Office Action (May 30, 1997), at 6) (emphasis added). As shown in the above prosecution statements, BIAX defined the term "variable number" as the variable value of the delay, which depends upon the circumstances of the order of the instructions. The delay value is explained in the '037 patent specification as follows:

> A ***delay field*** is built into the processing of instruction I5 so that even though it is processed in time interval T17 ***(the earliest possible time)***, its execution is delayed so that looping or branching out occurs after instruction I3 has executed.

(Ex. 1, col. 12, ll. 63-67).

Thus, based on the statements made by BIAX in the prosecution history and specification of the ’037 patent, the claim phrase “variable number of instruction cycles” should be construed to mean “a delay period lasting a number of instruction cycles that changes based on the assignment of the branch instruction to an earliest possible instruction firing time.”

BIAX’s position here – that this claim phrase does not need to be construed – ignores the prosecution history and the specification, and the prior art systems available before 1985. Again, it is against the prior art background – including the Schorr, Gross, and Young prior art – that the claims of the ‘037 patent must be construed. With that in mind, it is important to note that Schorr described a variable number of instructions scheduled between the “prepare-to-branch” and “exit” points. (Ex. 2, at 170); Young discussed the addition of a branch count field to the prepare-to-branch instruction which indicated when the branch would be executed, with the branch count shown to be a variable number. (Ex. 4, at 545, 547); and Gross described a delayed branch algorithm wherein non-branch instructions were moved to positions after the branch instruction, thereby rescheduling the branch instruction. (Ex. 3, at 116).

4. “Users” (claims 2, 4, 6, 9, 11, 12)

Phrase to be Construed	**ADI’s Construction**	**BIAX’s Construction**
“users”	Multiple persons using a computer system.	If the term “users” in the preamble is to be construed, it means persons or programs.

The term “users” appears in the preambles of asserted claims 2, 4, 6, 9, 11 and 12 of the ‘037 patent. ADI contends that “users” must be construed to be “multiple persons using a computer system.” BIAX contends that the word does not need to be construed, but if it is construed, it means “a person or a program.” Only ADI’s construction makes sense in light of the claim language, specifications and dictionary definitions.

The word “users” must be construed because the preambles of each of these claims provide antecedent basis for claim limitations recited in the body of the claims, thus making the preambles limiting. *See NTP, Inc. v. Research In Motion, Ltd.*, 418 F.3d 1282, 1306 (Fed. Cir.

2005) (holding that elements in the preamble limited the claim, since those limitations derived their antecedent basis from the preamble and were necessary to provide context for the claim limitations). The preamble to claims 2, 4, 6, 9, 11 and 12 of the '037 patent recite "in a plurality of programs utilized by a number of users." The body of these claims use the phrases "said programs," "each of said programs," and "all of said programs." To determine what program or programs are being referred to, one must consult the preamble. The preamble teaches that the programs being referred to are "programs utilized by a number of users," triggering the need to construe the word "users."

The word "user" cannot be construed to mean both a person or a program – as BIAX contends – because that would produce nonsensical claim language. Replacing the term "user" with the term "program" results in the nonsensical claim elements "plurality of ***programs*** utilized by a number of ***programs***" (emphasis added). Constructions that result in nonsensical claim language should be rejected. *See Schoenhaus v. Genesco, Inc*., 440 F.3d 1354, 1357 (Fed. Cir. 2006) (holding that plaintiff's proposed definition of disputed claim term would render claim nonsensical and was thus rejected).

The specification also makes it clear that a "user" is a person who uses a computer, not a program. For example, the specification states: "The present invention, therefore, pertains to a non-Von Neumann MIMD computer system capable of ***simultaneously operating upon many different and conventional programs by one or more different users***." (Ex. 1, col. 4, ll. 1-4). In addition: "The present invention can and preferably does support ***several users simultaneously*** in both time and space." (Ex. 1, col. 32, ll. 58-59). The specification further states that a "user" is capable of expending "effort, money and time" to prepare programs. (Ex. 1, col. 32, ll 25-27).

Finally, as set forth in Defendants' Joint Claim Construction Brief (*see* Section V.A.15), computer dictionaries from the period of time when the BIAX patent application was filed in 1985 defined "user" as a person or human being using a computer system. For example:

> **User:** "(1) Anyone who owns or utilizes a computer for problem solving or data manipulation. (2) Anyone who requires the services of a computer system."

(Ex. EE of Defs.' Joint Brief, *Webster's New World Dictionary of Computer Terms* 397 (1988)).

> **User:** "(1) Anyone who utilizes a computer for problem solving or data manipulation. (2) Anyone who requires the services of a computer system."

(Ex. Z of Defs.' Joint Brief, *Spencer's Computer Dictionary for Everyone* 279 (1985)).

> **User**: "Any person who uses a computer system or some part of one."

(Ex. AA of Defs.' Joint Brief, *The New American Dictionary of Computer Terms* 299 (1983)).

ADI's proposed construction of "user" as "multiple persons using a computer system" is fully supported by intrinsic and extrinsic evidence, and should be adopted.

5. "Beginning execution of said branch instruction" (claims 7, 8, 9, 12)

Phrase to be Construed	ADI's Construction	BIAX's Construction
"beginning execution of said branch instruction"	Loading the branch instruction into the instruction register of the branch execution unit.	Issuing the branch instruction.

The claim phrase "beginning execution of said branch instruction," which is in claims 7, 8, and 9 of the '037 patent,[14] should be construed to mean "loading the branch instruction into the instruction register of the branch execution unit." That definition gives the proper context to the claim phrase and is expressly set forth in the specification. *See Neomagic Corp. v. Trident Microsys, Inc.*, 287 F.3d 1062, 1071 (Fed. Cir. 2002) (looking to the specification "to improve our understanding" of claim language where "the language of the claim does not speak with clarity"). BIAX, however, proposes that this phrase should be construed as "issuing the branch instruction," which is more ambiguous than the claim language itself, and thus improper.

Fundamentally, ADI and BIAX appear to agree on ADI's construction, since BIAX admits in its opening brief that: "A branch instruction is 'issued' when it is sent to a branch execution unit and begins execution." BIAX's Br. at 23. Where the parties differ is whether "beginning execution" means "loading" (as ADI contends) or "sending" (as BIAX contends),

[14] Although the phrase "beginning execution of said branch instruction[s]" is contained within claims 3 and 4, the disputed language in those claims should be construed pursuant to 36 U.S.C. § 112, § 6. (*See* Section III.C.5).

and whether the branch instruction is loaded into the "instruction register of the branch execution unit" (as ADI contends) or just the "branch execution unit" itself (as BIAX contends).

BIAX's definition cannot mean "sending" the branch instruction, since BIAX's concedes that "beginning execution" does not happen until *after* the branch instruction is sent to the branch execution unit: "A branch instruction is 'issued' when it is sent to a branch execution unit and *begins execution*." BIAX's Br. at 23 (emphasis added). The act of "sending" does not begin execution of the branch instruction, but rather, it is the loading of loading the branch instruction into the instruction register of the branch execution unit that begins execution. (Ex. 1, col. 37, ll. 9-15; col. 38, ll. 49-54).

The specification of the '037 patent sets forth the meaning of this phrase. First, the specification is clear that: "***the branch execution unit (BEU) of the present invention executes all branch instructions***." (Ex. 1, col. 12, ll. 37-40) (emphasis added). Also: "***[T]he Branch Execution Unit (BEU) 1548 is the unit in the present invention responsible for the execution of all branch instructions which occur at the end of each basic block***." (Ex. 1, col. 36, ll. 20-23) (emphasis added). As the Federal Circuit has held, references to a particular feature as part of the "the present invention" – as is the case here – limit the claims to structures and methods utilizing that feature. *See Microsoft Corp.*, 357 F.3d at 1348.

In addition, the specification explains that "execution" of instructions is a process: "The present invention also makes extensive use of delayed-branching to guarantee program correctness. When a branch has executed and its effects are being propagated in the system, all instructions that are within the procedural domain of the branch must either have been executed *or be in the process of being executed*, as discussed in connection with the example of Table 6." (Ex. 1, col. 37, ll. 9-15) (emphasis added). In defining execution, the specification states: "In operation the branch execution unit (BEU) 1548 functions as follows. ***The branch instruction***, such as instruction I5 in the example above, ***is loaded into the instruction register*** 1900 from the PIQ bus interface 1544. The contents of the instruction register then control the further operation of BEU 1548." (Ex. 1, col. 38, ll. 49-54) (emphasis added). In addition: "[T]he branch

instruction is delivered over bus 1546 from the PIQ bus interface unit 1544 into the instruction register 1900 of the BEU [branch execution unit] 1548." (Ex. 1, col. 37, ll. 39-41).

As noted by the Federal Circuit, "[t]he construction that stays true to the claim language and most naturally aligns with the patent's description of the invention will be, in the end, the correct construction." *Phillips*, 415 F.3d at 1316. "Beginning execution of said branch instruction," should be construed to mean "loading the branch instruction into the instruction register of the branch execution unit."

6. "Completing the execution of said [scheduled] branch instruction" (claims 7-12)

Phrase to be Construed	**ADI's Construction**	**BIAX's Construction**
"completing the execution of said [scheduled] branch instruction"	Delivering the target address of a branch instruction to the instruction cache control so that the target instruction will be the next instruction fetched for execution.	Delivering the target address of a branch instruction so that the target instruction will be the next instruction fetched for execution.

BIAX has changed its construction of the claim phrase "completing the execution of said [scheduled] branch instruction"[15] (set forth in claims 7-12) to adopt ADI's proposed construction, except for leaving out where the target address is delivered to. The location of the delivery is important, however. This claim phrase should be properly construed as "delivering the target address of a branch instruction *to the instruction cache control* so that the target instruction will be the next instruction fetched for execution."

The specification of the '037 patent sets forth the meaning of this phrase:

> Hence, in the discussion of Table 6, instruction I5 is assigned a firing time T17 but is delayed until firing time T18. During the delay time, the interface 1950 signals the instruction cache control 1518 over line 1549 to continue to fetch instructions to finish the current basic block. ***When enabled, the interface unit 1950 delivers the next address (that is, the branch execution address) for the next basic block into the instruction cache control 1518 over lines 1549***.

[15] Although "completing the execution of said [scheduled] branch instruction" is contained within claims 1-6, the disputed language in those claims should be construed pursuant to 35 U.S.C. § 112, § 6. (*See* Section III.C.4-5).

> In summary and for the example on Table 6, the branch instruction I5 is loaded into the instruction register 1900 during time T17. However, a delay of one firing time (DELAY) is also loaded into the instruction register 1900 as the branch instruction cannot be executed until the last instruction I3 is processed during time T18. Hence, even though the instruction I5 is loaded in register 1900, the branch address for the next basic block, which is contained in the TARGET ADDRESS, does not become effective until the completion of time T18. In the meantime, the next instruction interface 1950 issues instructions to the cache control 1518 to continue processing the stream of instructions in the basic block. ***Upon the expiration of the delay, the interface 1950 is enabled, and the branch is executed by delivering the address of the next basic block to the instruction cache control 1518.***

(Ex. 1, col. 39, ll. 43-67).

"[T]he specification is always highly relevant to the claim construction analysis. Usually, it is dispositive; it is the single best guide to the meaning of a disputed term." *Vitronics*, 90 F.3d at 1582. Where, as here, the written description is not an alternative, but rather a description of the invention itself, that description necessarily controls the claim interpretation. *See Toro Co. v. White Consol. Indus, Inc.*, 199 F.3d 1295, 1301-02 (Fed. Cir. 1999). The claim phrase "completing the execution of said [scheduled] branch instruction" should be construed to mean "delivering the target address of a branch instruction to the instruction cache control so that the target instruction will be the next instruction fetched for execution."

7. "Time of execution" (claims 1-12) and "Time of execution of said branch instruction" (claims 1, 2, 5, 6, 10, 11, 12)

Phrases to be Construed	**ADI's Construction**	**BIAX's Construction**
"time of execution"	The time at which an instruction begins processing and execution.	When an instruction is issued.
"time of execution of said branch instruction"	The time at which a branch instruction begins processing and execution.	When the branch instruction is issued.

As BIAX notes in its opening claim construction brief, the parties' constructions of these terms is not far part. However, BIAX's constructions – "when an instruction *is issued*" and "when the branch instruction *is issued*" (emphasis added) – are more ambiguous than the claim language itself. BIAX appears to concede this point, since it offers a definition for "issued" in its brief: "An instruction is 'issued' when it is sent to a processor element and begins execution."

BIAX's Br. at 22-23. Using BIAX's explanation, the term "time of execution" cannot mean "when an instruction is issued," since BIAX concedes that execution does not begin until *after* an instruction is sent. Moreover, because the specification confirms that the execution of instructions is done by devices (*i.e.*, processor elements and branch execution units), the sending of an instruction cannot be the "time of execution" of the instruction: "It should be noted that the processor elements execute all non-branch instructions, while the branch execution unit (BEU) of the present invention executes all branch instructions." (Ex. 1, col. 12, ll. 37-40).

8. "Basic blocks" (claims 1-12)

Phrases to be Construed	**ADI's Construction**	**BIAX's Construction**
"single entry-single exit (SESE) basic blocks (BBs)" "basic blocks"	Compiler output consisting of blocks of instructions with only one branch at the end of each block.	"Basic blocks" means groups of contiguous instructions. "Single entry-single exit" has its plain and ordinary meaning: the block of instructions has a single entry point and a single exit point.

This term is fully argued in Defendant's Joint Claim Construction Brief filed herewith. The issues discussed in Section V.A.14 of the Joint Brief apply here as well, and therefore are incorporated by reference.

C. <u>Means-Plus-Function Limitations</u>.

Claims 1-6 of the '037 patent are written in "means-plus-function" format. BIAX and ADI agree that claims 1-6 contain means-plus-function elements that must be construed pursuant to 35 U.S.C § 112, paragraph 6, and agree on many of the claimed functions. However, the parties disagree on the corresponding structures that perform the functions set forth in these means-plus-function limitations.

The first step in construing a means-plus-function element is "to identify the function explicitly recited in the claim." *Asyst Technologies, Inc. v. Empak, Inc.*, 268 F.3d 1364, 1369 (Fed. Cir. 2001). The second step in construing a means-plus-function element is to "identify the corresponding structure set forth in the written description that performs the particular function set forth in the claim" as understood by one skilled in the art. *Id.*

In exchange for the ability of a patentee to use means expressions and avoid reciting in a patent claim "all possible structures that could be used as means," the patentee's claim must be limited to "the means specified in the written description and equivalents thereof." *Medical Instrumentation and Diagnostics Corp. v. Elekta AB*, 344 F.3d 1205, 1211 (Fed. Cir. 2003). Thus, the function is found by looking at the claim language itself, while the structure is found by looking at the specification. The corresponding structure must "actually perform the recited function, not merely enable the pertinent structure to operate as intended . . . ." *Asyst Technologies*, 268 F.3d at 1371. (ADI also incorporates by reference Section IV.E of the Defendants' Joint Brief, entitled "Means-Plus-Function Claims.")

Generally, asserted claims 1-6 of the '037 patent each have three means: (1) a means for determining the branch instruction and adding firing time information to the branch instruction (*see* Sections III.C.1 and III.C.2); (2) a means for processing non-branch instructions; (*see* Sections III.C.3); and (3) a means operative on the branch instruction in response to the firing time information, so that the execution of the branch instruction occurs in parallel with the execution of the non-branch instructions (*see* Sections III.C.4 and III.C.5).

1. Means for determining the branch instruction and for adding firing time information to said branch instruction (claims 1, 2, 5, 6)

Phrase to be Construed	**ADI's Construction**	**BIAX's Construction**
"means receptive of [each] said program for determining the branch instruction within each said basic block of said program[s], said determining means further adding firing time information to said branch instruction[s]," (claims 1 and 2) "means receptive of [each]said program for determining the branch instruction within each said basic block of said program[s], said determining means further adding instruction firing time information to said scheduled branch	The claimed functions are "determining the branch instruction within each said basic block of said program[s]" and "adding firing time information to said branch instruction" (claims 1 and 2); and "determining the branch instruction within each said basic block of said program[s]" and "adding instruction firing time information to said scheduled branch instruction" (claims 5 and 6). The corresponding structure is the computer processing system 160 and the TOLL software algorithms disclosed in the specification, which	This function does not require construction because certain words ("determining") are clear on their face and other terms are construed elsewhere. If "branch instruction" needs to be construed, see proposed construction in Section VI.B.18. If "firing time information" needs to be construed, see Section VI.B.25. If "basic block" needs to be construed, see Section VI.B.1. (Sections refer to BIAX's Opening Brief). The corresponding structure is shown in FIG. 1: computer processing system 160; TOLL

instruction[s]," (claims 5 and 6)	operate on compiled output that adds to each of the branch instructions computer code that assigns an instruction firing time to the branch instructions, as shown in Fig. 4, 7a-7c, 8, 10, stage 1040; Tables 4-6, 10, 14, 16, 17; and described in the '037 patent, 4:37-62; 6:33-7:35; 8:39-10:1; 10:43-50; 11:16-13:8; 13:30-39; 15:48-67; 16:29-38; 16:46-47; 18:27-23:32; 30:38-60; 31:32-47; 31:65-32:48; 33:46-49; 34:1-9; 39:43-67; 41:5-45:35.	software 110.

The parties agree that the functions of the "means . . . for determining the branch instruction" are, for claims 1 and 2, "determining the branch instruction within each said basic block of said program[s]" and "adding firing time information to said branch instruction;" and for claims 5 and 6, "determining the branch instruction within each said basic block of said program[s]" and "adding instruction firing time information to said scheduled branch instruction."

It is undisputed that the corresponding structure to the above functions includes at least the TOLL software 110 operating on computer processing system 160. BIAX Br. at 39. However, under the Federal Circuit's decision in *WMS Gaming Inc. v. Int'l Game Tech.,* 184 F.3d 1339, 1349 (Fed. Cir. 1999), the specific software algorithms disclosed in the specification must be identified. As set forth in *WMS Gaming*, where the structure corresponding to a "means-plus-function" limitation is a computer, the proper construction is not simply a computer, but rather a computer programmed to perform the specific algorithms disclosed in the specification. *WMS Gaming*, at 1349. *See also Harris Corp. v. Ericsson Inc.*, 417 F.3d 1241, 1253 (Fed. Cir. 2005) ("[T]he court in *WMS Gaming* rejected the argument that the corresponding structure was merely 'an algorithm executed by a computer,' holding instead that it was limited to the specific algorithm disclosed in the specification."); *Gobeli Research Ltd v. Apple Computer, Inc. & Sun Microsystems, Inc.*, 384 F. Supp. 2d 1016, 1023 (E.D. Tex. 2005) (holding claim invalid due to failure to disclose software algorithm in specification). Because BIAX's construction does not

include an identification of any software algorithms performed on computer processing system 160, BIAX's construction cannot be adopted.

The proper structure for the claimed functions is the computer processing system 160 and the TOLL software algorithms disclosed in the specification, which operate on compiled output that adds to each of the branch instructions computer code that assigns an instruction firing time to the branch instructions, as shown in Fig. 4, 7a-7c, 8, 10, stage 1040; Tables 4-6, 10, 14, 16, 17; and described in the '037 patent, 4:37-62; 6:33-7:35; 8:39-10:1; 10:43-50; 11:16-13:8; 13:30-39; 15:48-67; 16:29-38; 16:46-47; 18:27-23:32; 30:38-60; 31:32-47; 31:65-32:48; 33:46-49; 34:1-9; 39:43-67; 41:5-45:35.

2. Means for determining the branch instruction and for scheduling processing of branch instructions (claims 3, 4)

Phrase to be Construed	**ADI's Construction**	**BIAX's Construction**
"means receptive of [each] said program[s] for determining the branch instruction within each said basic block of [each of] said program[s], said determining means further scheduling processing of said branch instruction[s],"	The claimed functions are: (1) determining the branch instruction and (2) adding an instruction firing time to the branch instruction. The corresponding structure is: the computer processing system 160 and the TOLL software algorithms disclosed in the specification, which operate on compiled output that adds to each of the branch instructions computer code that assigns an instruction firing time to the branch instructions, as shown in Fig. 4, 7a-7c, 8, 10, stage 1040; Tables 4-6, 10, 14, 16, 17; and described in the '037 patent, 4:37-62; 6:33-7:35; 8:39-10:1; 10:43-50; 11:16-13:8; 13:30-39; 15:48-67; 16:29-38; 16:46-47; 18:27-23:32; 30:38-60; 31:32-47; 31:65-32:48; 33:46-49; 34:1-9; 39:43-67; 41:5-45:35.	This term does not require construction because certain words ("determining") have a clear meaning, and other terms ("basic block") are construed elsewhere. The corresponding structure is shown in Fig. 1: computer processing system 160; TOLL software 110.

The parties agree that the functions performed by the "means . . . for determining the branch instruction" are "determining the branch instruction within each said basic block of [each of] said program[s]" and "scheduling processing of said branch instruction[s]."

It is undisputed that the corresponding structure to the above functions includes at least the TOLL software 110 operating on computer processing system 160. BIAX Br. at 39. However, because BIAX's construction does not include an identification of any software algorithms performed on computer processing system 160, BIAX's construction cannot be adopted under Federal Circuit precedent. *WMS Gaming*, 184 F.3d at 1349.

The limitation above calls for scheduling processing of branch instructions firing time, while the corresponding limitations from claims 1, 2, 5, and 6 of the '037 patent call for adding firing timing information. ADI contends that in the context of these mean-plus-function claim elements, scheduling processing of branch instructions must mean adding firing timing information. Indeed, if the phrase "scheduling processing of branch instructions" were to be given a different construction, claims 3 and 4 would be invalid for lack of a written description. *See, e.g., The Gentry Gallery, Inc. v. The Berkline Corp.*, 134 F.3d 1473, 1479 (Fed. Cir. 1998); *see also Gobeli Research*, 384 F. Supp. 2d at 1023. This is because the *only* disclosure in the specification of a software algorithm adding timing information are algorithms relating to an instruction firing time which operates on the instruction stream after compilation and prior to execution. The scope of these means-plus-function claim elements, therefore, must be limited to the embodiment disclosed in the specification, and equivalents. *See Signtech USA, Ltd. v. Vutek, Inc.*, 174 F.3d 1352, 1356 (Fed. Cir. 1999) (holding that where a specification disclosed "little more than the preferred embodiment," the structure for a means-plus-function element was limited to the structure of the preferred embodiment); *Fonar Corp. v. General Elec. Co.*, 107 F.3d 1543, 1551-52 (Fed. Cir. 1997) (holding that a means-plus-function element is limited to the specifically disclosed "generic gradient wave forms," despite the specification's disclosure that "other wave forms may be used," because that general disclosure lacked specificity).

The proper structure for the claimed functions is the computer processing system 160 and the TOLL software algorithms disclosed in the specification, which operate on compiled output that adds to each of the branch instructions computer code that assigns an instruction firing time to the branch instructions, as shown in Fig. 4, 7a-7c, 8, 10, stage 1040; Tables 4-6, 10, 14, 16, 17; and described in the '037 patent, 4:37-62; 6:33-7:35; 8:39-10:1; 10:43-50; 11:16-13:8; 13:30-39; 15:48-67; 16:29-38; 16:46-47; 18:27-23:32; 30:38-60; 31:32-47; 31:65-32:48; 33:46-49; 34:1-9; 39:43-67; 41:5-45:35.

3. Means for processing non-branch instructions (claims 1-6)

Phrase to be Construed	**ADI's Construction**	**BIAX's Construction**
"means operative on the received non-branch instructions in each said basic block for processing said instructions" (claims 1, 3, 5) "means operative on the received non-branch instructions in each said basic block of each said program for processing said programs" (claims 2, 4, 6)	The claimed function is processing non-branch instructions. The corresponding structure is Processor Elements (PEs) 640 which are identical, homogenous, context-free devices that each can execute all instructions except branch instructions, as shown in Figs. 6, 7a-7c, 15, 18 and 21, and described in the '037 patent, 12:31-67; 13:30-42; 14:24-53; 15:9-27; and 40:20-45:33.	Does not require construction because it has the ordinary meaning of the words in the term and any terms needing construction are construed elsewhere herein. If "branch instruction" needs to be construed, see Section VI.B.18. If "basic block" needs to be construed, see Section VI.B.1. (Sections refer to BIAX's Opening Brief). The corresponding structure is shown in the '037 Patent at FIG. 6, PEs 640.

BIAX agrees that ADI accurately describes the functions claimed by then above phrases. *See* BIAX Br. at 42. BIAX also agrees that the corresponding structure is processor elements (PEs) 640 as described in the specification. *Id*. BIAX, however, objects to ADI's citation to portions of the specification that describe the structure as "identical" and "context free" processor elements. *Id*. BIAX's objections are without merit.

The '037 patent specification makes it clear that the processor elements as disclosed are both ***identical*** to each other and are ***context-free*** (*i.e*. do not retain execution state information). For example, the specification states:

> The results of this static allocation are executed on a system containing a plurality of processor elements. In one embodiment of the invention, ***the processors are identical***. The processor elements, in this illustrated embodiment, contain no execution state information from the execution of previous instructions, that is, ***they are context free***.

(Ex. 1, col. 4, ll. 15-21) (emphasis added).

Indeed, the original specification, unequivocally referred to the processor elements as "identical":

> ***The results of this static allocation are executed on a system containing a plurality of identical processor elements***. These processor elements are characterized by the fact that they contain no execution state information from the execution of previous instructions, i.e., ***they are context free***.

(Ex. 2, Original Specification, col. 4, ll. 11-16) (emphasis added).

> In FIG. 21, the details of the processor elements 640 are set forth for a four-stage pipeline processor element. ***All processor elements 640 are identical***. It is to be expressly understood, that any prior art type of processor element such as a micro-processor or other pipeline architecture could not be used under the teachings of the present invention ***because such processors retain the state information of the program they are processing***.

(Ex. 2, Original Specification, col. 48, ll. 25-34) (emphasis added).

Even in the substitute specification, in which BIAX attempted to systematically remove the references to "identical" processor elements, the *only* disclosed embodiment described consists of identical and context-free processor elements for processing instructions:

> Referring to FIG. 21, a particular processor element 640 has a four-stage pipeline processor element. ***All processor elements 640 according to the illustrated embodiment are identical***. It is to be expressly understood , that any prior art type of processor element such as a micro-processor or other pipeline architecture could not be used under the teachings of the present invention, ***because such processors retain substantial state information of the program they are processing***.

(Ex. 1, col. 41, ll. 5-13) (emphasis added).

> Each processor element is homogeneous and capable, by itself, of executing all computational and data memory accessing instructions.

(Ex. 1, col. 41, ll. 22-24) (emphasis added). (*See also* Ex. 1, col. 4, ll. 17-28; col. 15, ll. 9-24).

Further, Figure 6 shows that the processor elements are identical, by indicating, using the nomenclature PE_0 followed by a vertical ellipses and then followed by PE_k, that there is an arbitrary number (k) of the same processor element shown in this figure.

Regardless of whether the disclosed embodiment in the specification is characterized as the entire invention, or as an embodiment of the invention, it is the ***only*** embodiment disclosed in the specification. The specification does not describe any other embodiment that does not have identical and context free processor elements. The scope of these means-plus-function claim elements must be limited to the embodiment disclosed in the '037 patent specification, and equivalents. *See Signtech*, 174 F.3d at 1356; *Fonar*, 107 F.3d at 1551-52.

4. Means for completing execution of said branch instruction (claims 1, 2, 5, 6)

Phrase to be Construed	**ADI's Construction**	**BIAX's Construction**
"means operative on said received branch instruction in said basic block in response to the firing time information for completing the execution of said branch instruction no later than the same time as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block" (claims 1, 2) "means operative on said received branch instruction in said basic block in response to said time information, for completing the execution of said scheduled branch instruction no later than during the same time as said processing means is processing the last to be executed non-branch instructions in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions"	The claimed function is completing the execution of branch instructions no later than the same time as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of the branch instruction occurs in parallel with the execution of the non-branch instructions in the basic block. The corresponding structure is a Delay Unit 1940 in a Branch Execution Unit 1548, containing a delay field in an Instruction Register 1900, and connected to the Instruction Cache Control Unit 1518, that causes final execution of the branch instruction to be delayed based on a delay value in a delay field in a register of the computer, as shown in Figs. 15, 16, 17, and 19; and described in the '037 patent, 12:18-20; 12:32-67, Table 6; 36:19-40:18; 41:5-	Does not require construction because it has the ordinary meaning of the words in the term and any terms needing construction are construed elsewhere herein. The corresponding structure is shown in the shown in the '037 Patent at FIG. 19: Branch Execution Unit (BEU) 1548; Delay Unit 1940.

(claim 5) "means operative on said received branch instructions in each said basic block for completing the execution of said scheduled branch instruction no later than during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block for a given program so that the executed of said branch instructions occurs in parallel with the execution of said instructions in said basic block" (claim 6)	45:32.	

The parties agree that the function performed by the "means . . . for completing execution of said branch instruction" are "completing the execution of said branch instruction no later than the same time as said processing means is processing the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block" (claims 1 and 2); "completing the execution of said scheduled branch instruction no later than during the same time as said processing means is processing the last to be executed non-branch instructions in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions" (claim 5); and "completing the execution of said scheduled branch instruction no later than during the same time as said processing means is processing the last to be executed non-branch instruction in said basic block for a given program so that the execution of said branch instructions occurs in parallel with the execution of said instructions in said basic block" (claim 6).

It is undisputed that the corresponding structure to the above functions includes at least the Branch Execution Unit 1548 and the Branch Delay Unit 1940. BIAX Br. at 43. The proper corresponding structure, however, not only includes a Delay Unit 1940 in a Branch Execution Unit 1548, but also a delay field in an Instruction Register 1900, connected to the Instruction Cache Control Unit 1518, that causes final execution of the branch instruction to be delayed

based on a delay value in a delay field in a register of the computer, as shown in Figs. 15, 16, 17, and 19; and described in the '037 patent, 12:18-20; 12:32-67, Table 6; 36:19-40:18; 41:5-45:32.

BIAX dictated the correct claim scope, and its intention, as expressed in the specification, should be regarded as dispositive. *See SciMed Life Sys., Inc. v. Advanced Cardiovascular Sys., Inc.*, 242 F.3d 1337, 1343-44 (Fed. Cir. 2001).

5. Means for beginning execution of said branch instruction and for completing execution of said branch instruction (claims 3, 4)

Phrase to be Construed	**ADI's Construction**	**BIAX's Construction**
"means operative on said received branch instruction in said basic block for beginning execution of said branch instruction at one of a variable number of instruction cycles prior to a time of execution of a last to be executed instruction of said basic block and for completing the execution of said scheduled branch instruction during an instruction cycle which occurs no later than during the processing of the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block" (claims 3, 4)	The claimed function is: (1) beginning execution of said branch instruction at one of a variable number of instruction cycles prior to a time of execution of a last to be executed instruction of said basic block, and (2) completing the execution of said scheduled branch instruction during an instruction cycle which occurs no later than during the processing of the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block The corresponding structure is a branch execution unit 1548 containing: (1) an instruction register 1900 that loads the branch instruction according to its added firing time information into the instruction register 1900 from the instruction cache control unit 1518 via the bus interface unit 1544 at one of a variable number of instruction cycles prior to a time of execution of a last to be executed instruction of said basic block; and (2) a delay unit 1940 containing a delay field in an instruction register 1900, and connected to the instruction cache control unit 1518, that causes final execution of the branch instruction	Does not require construction because it has the ordinary meaning of the words in the term and any terms needing construction are construed elsewhere herein. If "branch instruction" needs to be construed, see Section VI.B.18. If "firing time information" needs to be construed, see Section VI.B.25. If "basic block" needs to be construed, see Section VI.B.1. If "instruction cycle" needs to be construed, see Section VI.B.26. The corresponding structure is Fig. 19: BEU 1548. The corresponding structure is shown in the shown in the '037 patent at Fig. 19: Branch Execution Unit (BEU) 1548; Delay Unit 1940.

	to be delayed based on a delay value in the delay field, as shown in Figs. 15, 16, 17, and 19; and described in the '037 patent, 12:18-20; 12:32-67, Table 6; 36:19-40:18; 41:5-45:32.	

The function performed by the above means is two-fold: (1) "beginning execution of said branch instruction at one of a variable number of instruction cycles prior to a time of execution of a last to be executed instruction of said basic block"; and (2) "completing the execution of said scheduled branch instruction during an instruction cycle which occurs no later than during the processing of the last to be executed non-branch instruction in said basic block so that the execution of said branch instruction occurs in parallel with the execution of said non-branch instructions in said basic block."

It is undisputed that the corresponding structure to the above functions includes at least the Branch Execution Unit 1548 and the Branch Delay Unit 1940. BIAX Br. at 43. The proper corresponding structure, however, includes a Branch Execution Unit 1548 containing: (1) an instruction register 1900 that loads the branch instruction according to its added firing time information into the instruction register 1900 from the instruction cache control unit 1518 via the bus interface unit 1544 at one of a variable number of instruction cycles prior to a time of execution of a last to be executed instruction of said basic block; and (2) a delay unit 1940 containing a delay field in an instruction register 1900, and connected to the instruction cache control unit 1518, that causes final execution of the branch instruction to be delayed based on a delay value in the delay field, as shown in Figs. 15, 16, 17, and 19; and described in the '037 patent, 12:18-20; 12:32-67, Table 6; 36:19-40:18; 41:5-45:32.

IV. CONCLUSION

For the foregoing reasons, ADI respectfully requests that this Court construe, as a matter of law, the claim language as set forth above.

Dated: October 30, 2006

Respectfully submitted,

By: _____

David J. Beck, *Lead Attorney*
Texas Bar No. 00000070
Beck, Redden & Secrest, LLP
One Houston Center
1221 McKinney St., Suite 4500
Houston, TX 77010
Tel. (713) 951-3700
Fax (713) 951-3720
dbeck@brsfirm.com

ATTORNEY-IN-CHARGE FOR DEFENDANT ANALOG DEVICES, INC.

Of Counsel:
Robert D. Daniel
Jim Taylor
Beck, Redden & Secrest, LLP
One Houston Center
1221 McKinney St., Suite 4500
Houston, TX 77010
Tel. (713) 951-3700
Fax (713) 951-3720

Jennifer Parker Ainsworth
Wilson, Sheehy, Knowles, Robertson & Cornelius P.C.
909 ESE Loop 323, Suite 400
Tyler, TX 75701
Tel. (903) 509-5000
Fax (903) 509-5092

Steven M. Bauer
Jeremy P. Oczek
Proskauer Rose LLP
One International Place
Boston, MA 02210
Tel. (617) 526-9600
Fax (617) 526-9899

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of the foregoing document was served in compliance with the Federal Rules of Civil Procedure and Local Rules of this District via E-Mail on this 30th day of October, 2006 to counsel of record.

Jim Taylor